Open Ethernet-Based Embedded Platforms for Integrated Modular Architectures and Autonomous Systems

Mirko Jakovljevic
Solution Architect – Critical Integrated Systems
TTTech Computertechnik AG
Vienna, Austria

Alvaro Soares
Principal Solutions Architect – Commercial & Military Aerospace
TTTech North America
Andover, MA, USA

ABSTRACT

By adopting the latest developments from other critical (e.g. integrated modular avionics) and high-volume automotive industries with safety requirements (ADAS and autonomous driving), the rotorcraft industry could reduce system lifecycle costs and gain new integrated platform capabilities which support incremental modernization, simplify upgrades and modifications for different missions or rotorcraft platforms. A specific set of architecture design patterns and computational models, used in integrated modular architectures, enables the design of less complex integrated systems which can collect and process all system sensor data in (hard) real-time, supports seamless sensor data fusion for IVHM, and enables the integration of critical and non-critical functions. Accompanied with robust system engineering, RTCA DO-254 / DO-178C DAL A/B design assurance and extended use of ASIL-D-compliant (automotive) components, novel integrated architectures for rotorcraft can be designed to fit with robust modular form factors such as VPX. Such integrated architectures can be extended with COTS computing and sensor fusion LRUs/ECUs used for automotive ADAS/ADS (advanced driver assistance systems/autonomous driver systems) and rapidly progressing autonomous driving applications.

NOTATION / ACRONYMS

ADAS	Advanced Driver Assistance Systems
ADS	Autonomous Driver Systems
AFDX	Avionics Full Duplex Switched Ethernet
ASIL	Automotive Safety Integrity Level
COTS	Commercial off the Shelf
DAL	Design Assurance Level
DSP	Digital Signal Processor
ECU	Electronic Control Unit
IMA	Integrated Modular Architecture (or Avionics)
IVHM	Integrated Vehicle Health Management
LRU	Line Replaceable Unit
OEM	Original Equipment Manufacturer
RT	Real Time
SBC	Single Board Computer
TDMA	Time Division Multiple Access
TSN	Time Sensitive Networking
TTA	Time Triggered Architecture
VL	Virtual Link

INTRODUCTION

The trend towards higher levels of functional integration, data fusion, more stringent safety requirements and standards in autonomous, automotive and manufacturing systems has erupted since 2010. This creates new high-volume applications for modular products and embedded platform components which are valued in critical embedded and infrastructure applications. New emerging applications such as ADAS (advanced driver assistance systems) / autonomous driving and functional system safety standards in automotive industry (e.g. ISO26262) drive the interest of semiconductor industries, and may allow low-volume industries such as aerospace to take advantage of commercial COTS products and technology roadmaps, which were previously not suited for critical applications.

Increasingly, advanced commercial aerospace methodology and experience in complexity management in integration of many sensors and computing units for IMA (Integrated Modular Architectures) is being leveraged to support highly integrated automotive and high-end industrial applications (Ref 1). In certain scenarios, there is a potential for converging requirements and sharing of key architecture

Prepared for AHS International 73rd Annual Forum & Technology Display, Fort Worth, Texas, USA, May 9-11, 2017.

patterns in design. However, system architectures will likely differ in topology, complexity, and node count between large commercial avionics and automotive vehicles. Passenger cars are designed for up to 10,000 hours in operation (Ref. 2,3), while commercial aircraft have a useful lifetime of 100,000 or more hours. In the rotorcraft domain, lifespans can range up to 50,000 hours, with similar redundancy, availability, integrity and reliability requirements to commercial aviation.

Recognizing that key differences exist in these industry spaces, an ideal solution would contain architectural patterns that allow for the definition of a scalable complex architecture based on a few LRUs with computing, Ethernet switching, safe IO and payload integration. Also, the integration of high-end computing modules (i.e. ECU/LRU) which support safe and autonomous operation at up to 15,000 feet, -40 to 85°C and 3-10G may simplify future sensor fusion and functional upgrades using innovations from other high-volume industries (i.e. automotive).

A generic redundant architecture which can integrate mission systems with different safety-, mission-, and time-critical functions may be built by using VPX-based networked architectures, and it can tightly integrate external COTS LRU/ECUs into the system by using deterministic Ethernet network. Another option would be to implement high-performance ADAS unit designs in VPX SBC (see Chapter "Integrating COTS Hardware For Autonomous Operation").

To be able to do so, some essential technology baseline must be deployed to enable integration of many distributed functions with different criticality levels.

OPEN INTEGRATED ARCHITECTURES

To define open integrated architecture, a few challenges shall be solved at once. Modularity shall be supported at hardware/equipment level, software platform level including software abstraction COM layers, and system integration level. At once the platform shall support different computation models and architectural patterns (mandatory for design of less complex integrated architectures), different topologies (more or less de-centralized), and the system behavior and performance ideally should be defined by parameters. Furthermore, key system interfaces between applications and components shall be also defined by parameters, and not by the application activity. By doing so we can reuse a set of components configurable for different applications.

Distributed platform hardware will consist of few part numbers – generic components which can be reprogrammed and reconfigured by parameters, to accommodate different platforms and missions. The component and equipment list will be limited to the following part classes (see Figure 1):

- **Core computing resources** (e.g. double or triple redundant) – high-end computing units with high-integrity backbone connectivity and single-board computer connectivity. Each SBC LRM can will host typically 3-10 (or more) applications.
- **LRU Switching Unit** – standalone deterministic Ethernet switch
- **LRM Switching Unit** – deterministic Ethernet switch VPX module
- **RDC** – Remote Data Concentrators with limited processing power, and a number of different databus and IO interfaces MIL-1553 / Fibre Channel / ARINC429 / CAN/Firewire / RS485 / RS 422 / RS232 / Analog & Digital IO / etc. are versatile interfacing units to connect with legacy equipment or host high-speed control loops closer to controlled objects
- **RIO Smart IO/Sensor/Actuator/Payload interface** – more specialized interfaces for specific classes and sensors, actuators and payload types. They can include more or less powerful computing depending on the sensor/payload.

In some cases, RDC and switching units can be placed together to form distributed common computing and switching resources (**dCCSR**) LRU, to improve reconfigurability and survivability, and reduce SWaP consumption. If carefully designed, such distributed units can play multiple roles and replace several other types of equipment.

System integration portion is configured by using the switching components described above and additional software abstraction middleware layers which are hosted on a software platform. If those software and hardware components are designed to support safety assurance design in line with DAL-A DO-178C/DO-254 and support DO-297 and the system has been designed with ARP4754/4761 as a part of the process, fulfill DO-160G Class B2 of systems, then it is possible to design infrastructure which can satisfy different commercial and rotorcraft requirements.

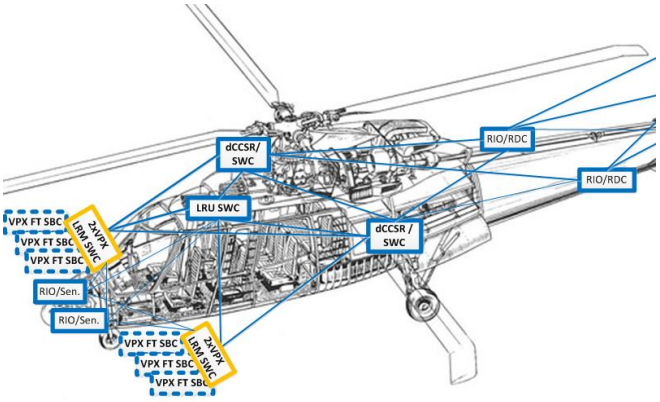

Figure 1. Example: Integrated architecture components and generic topology

VPX FOOTPRINT FOR OPEN INTEGRATED ARCHITECTURES

Different variants of generic open architectures can be implemented by using VPX-based Ethernet backplane and backbone networks, assuming they provide absolute temporal guarantees, congestion-free communication and determinism for different critical functions (Figure 2). The connectivity and topology of such an architecture can be tailored to satisfy different rotorcraft OEM platform optimization criteria.

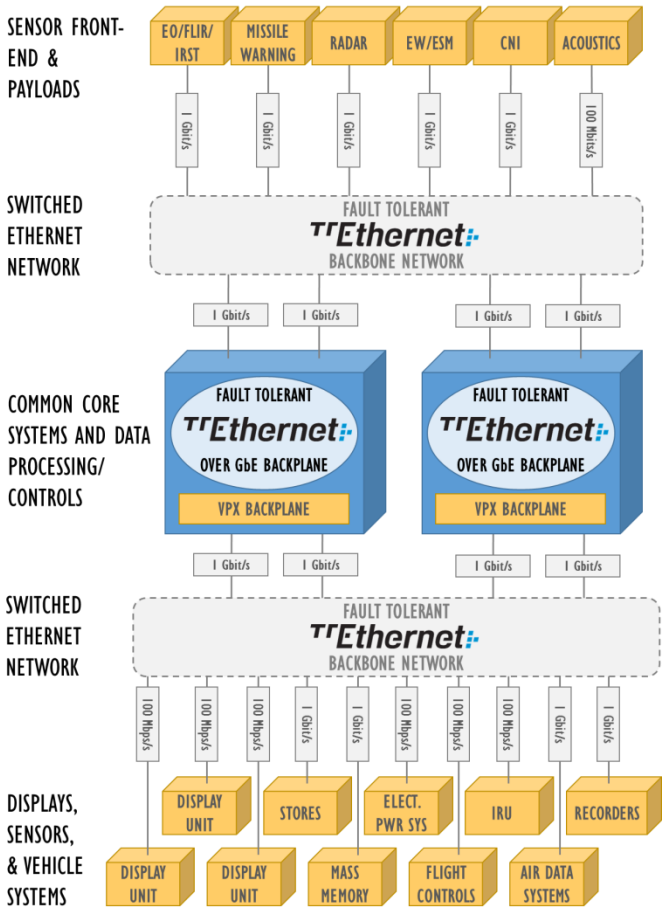

Figure 2. Integrated redundant architecture with two modular VPX computers, deterministic Ethernet backplane (2x LRM Switching Unit per Rack), and reconfigurable redundant payload and vehicle system connectivity.

DETERMINISTIC HIGH-BANDWIDTH ETHERNET BACKBONE

So called control-plane applications in VPX typically use single- or dual-star (redundant) topology with switched Gigabit Ethernet, which are also supported by VPX switches with SAE AS6802 (Ref. 4) Quality of Service (TTEthernet switch). Depending on the application, deterministic Ethernet switches can be used for control plane, data plane, and some utility plane applications (synchronization) in VPX-based systems. This means all functions and modules connected to backplane and backbone networks operate as if they are connected directly to a large, fault-tolerant Ethernet backbone. By allowing robust TDMA partitioning of networking resources, the system designer can determine the level of integration/interaction or isolation among different functions. This enables the design of innovative architectures and distributed platforms that can host many distributed functions using shared computing/ networking resources for advanced integrated system architectures. Furthermore, deterministic Ethernet switches can integrate ARINC664 (Ref.5) traffic class (see Figure 3) or any other IEEE802 QoS service.

Figure 3. Core switching VPX LRM designed for DAL A DO-254/DO-178C/SAE ARP 4754/4761

From the perspective of applications, the integrated architecture is integrated by a “virtual bus” and completely abstracted from the underlying topology. The temporal behavior of different data streams and network resource sharing for hard RT functions can be defined by parameters. It is possible to define robust key system interfaces and prevent congestions in the system, by utilizing selective policing, shaping, and time-triggered packet-forwarding policies for all critical data streams. With additional software platform mechanisms and inter-layer alignment (synchronization), system architects can define non-blocking models of computation which support predictable integration, isolation and reconfigurable interaction of different functions in the system.

The core switch modules and end station network interface cards are designed using safety assurance for the highest criticality levels (DAL A), using DO-254/DO-178C and SAE ARP4754/4761 considerations.

ARCHITECTURE PATERNS FOR ADVANCED INTEGRATED SYSTEMS

The framework for scalable embedded platforms in advanced integrated systems can include commercially available COTS computing/DSP hardware, and at the same time:

- Enable tight hard real-time control function coupling with a priori defined interfaces and separation of temporal and functional behavior, with simplified obsolescence management
- Support full isolation, and partitioning of hosted distributed functions per design,
- Support fully integrated and/or simple point-to-point integration, with high bandwidth and low bandwidth interfacing by using a blend of commercial system integration capabilities and standard avionics networks.

Primarily, this can be accomplished by using an appropriate framework for defining the embedded platform and its layer alignment. Models of computation and communication are of utmost importance in design of advanced integrated architectures, and their key properties are briefly presented.

MODELS OF COMPUTATION AND COMMUNICATION (MOCC)

As described in Ref 6., in Loosely Time Triggered Architecture (L-TTA) computing model (Ref 7), the application layer is decoupled and not synchronized to the periodic network communication with defined maximum latency.

The application is periodically repeated with given cycle based on local timer, and does not have any blocking or time-dependent behavior for used networking and computing resources. In conjunction with deterministic communication with defined maximum latency, system architects can define accurate control loop behavior with deterministic IO resource access.

In TTA (Ref 8), the execution of tasks and partitions is tied to system time, and synchronized with periodic TDMA network operation. This enables fixed I/O data transfer latency and system-level partitioning (Ref 9). This computational model is not affected by the distance among I/O and software functions or their spatial proximity. Different distributed tasks receive and send data periodically just in time for the processing based on predefined time-driven schedule.

The relaxed variant of TTA, which does not synchronize local clocks to system/network time, can be seen as a L-TTA model because the latency varies depending on the difference between the application sending time instant and the next time instant for transmission over the network. This approach supports fixed latency for network communication, and supports defined maximum end-to-end messaging latency. The perspective on MoCC capabilities is summarized in (Ref 10) and Figure 4. For embedded cloud computing, a mixed MoCC shall be used to support different real-time applications.

MoCC	Network Devices	Computing Modules	Supported by:
TTA: Time-Triggered Architecture (hard RT)	Synchronized to system time – fixed comm. latency	Synchronized to syst. time, time partition or task exec. scheduled to system time	Synchronous VLs and time-partitioning
Relaxed TTA (RT)	Synch. to system time – fixed comm. latency	Not synch. to system time, cyclic, rate monotonic task or partition exec.	Synchronous VLs and time-partitioning
Loosely TTA or L-TTA (RT)	Not synch. – defined max. latency with varying jitter	Not synch., cyclic, rate monotonic task or partition execution	Asynchronous VLs and time-partitioning
Mixed MoCC= TTA+L-TTA (hard RT and RT)	Both synch. and non-synch. – fixed latency or defined max. latency	Both synch. and non-synch. in integrated architectures	Asynch/Synch VLs and time-partitioning, system-level-time partitioning

Figure 4. Comparing MoCC, computing/networking layer alignment and performance (adapted from Ref. 10)

EMBEDDED VIRTUALIZATION FOR INTEGRATED SYSTEMS

EMBEDDED SYSTEM VIRTUALIZATION: STRUCTURAL VIEW – SYSTEM AS A COLLECTION OF PARTITIONS

The simplest system-level abstraction of a system is to consider it as a collection of many software partitions with their own hardware resources, which all together host many distributed functions. One or more partitions can host one localized or distributed function. When all those partitions are properly assigned (according to zonal safety, survivability, installation, environmental and other requirements) and installed on computers connected to the virtual bus, they build an integrated real-time system.

One computer can host one or more partitions separated in time and space. Simpler units (smart sensors, actuators, IO, etc.) hosting only one function can be seen as one physically separated partition hosted on one computer. In a networked system, all those partitions have well-defined key system interfaces and temporal interactions (e.g. periodic exchange of state variables) via virtual bus, while all less critical functions can use the remaining computing and networking bandwidth.

Such an approach ensures that the system architect can modify or upgrade functions, without modification and

additional verification of already installed and tested functions. However, to properly achieve performance goals, specific architecture patterns shall be carefully implemented.

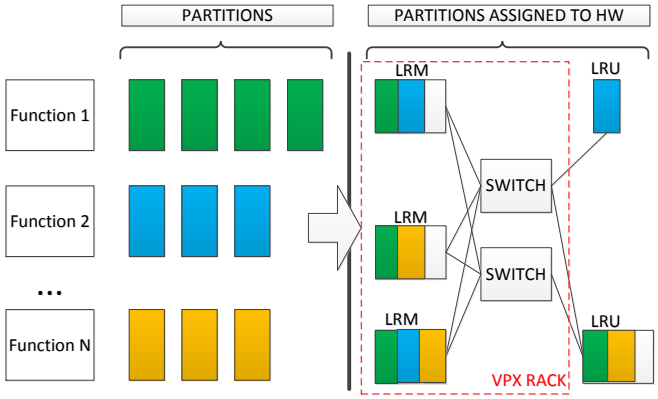

Figure 5. Partitions and their assignment to physical computers networked by deterministic Ethernet

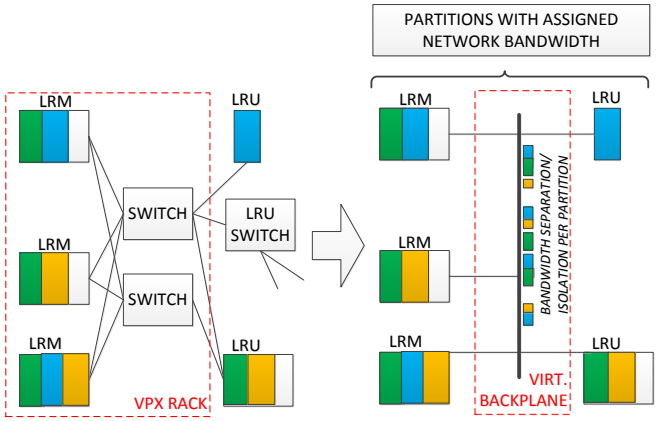

Figure 6. Deterministic Ethernet as virtual bus

SYSTEM INTEGRATION VIRTUALIZATION

DETERMINISTIC ETHERNET AS VIRTUAL BUS

In deterministic Gigabit-Ethernet networks, it is possible to emulate reflective memory by using a periodic global data exchange with applications that are synchronized to the global time base (e.g. generated either at the network level by fault-tolerant distributed clocking defined in SAE AS6802 services, or distributed from some central resources as defined in IEEE1588). From the logical perspective, different distributed functions gain a private, configurable congestion-free shared memory. The reconfiguration allows to group data of specific sets of functions and isolate from other functions, or it allows the emulation of separate logical circuits (unidirectional deterministic logical links, or virtual links which can operate as unicasts or multicasts).

By using this approach, we can scale the level of functional integration without influencing other existing functions in the system. Also, distributed applications do not need to know about underlying architecture or topology.

Therefore, sensor fusion and distributed payload processing can be executed without fear of unintended interactions with other system functions. Voting on data from synchronous sources simplifies redundancy management and application software design. Obsolescence management, modernization, and upgrades with new DSP processors and applications are simplified, as the behavior of already integrated functions will not change and cause new system integration or timing issues. Critical, hard real-time functions will not be influenced by other less critical distributed functions. Sensor front-end data can be streamed to platform systems or common core computing systems, with exact latency and no jitter, independent of network load. This also means that processing functions do not require spatial proximity to a specific sensor or controlled object, and can be placed anywhere in the system. This also simplifies reconfiguration, upgrades, and incremental modernization.

DETERMINISTIC ETHERNET PROTOCOLS AND SERVICES

In commercial avionics, ARINC664-P7 (Ref 5) (Avionics Full Duplex Ethernet or AFDX) QoS enhancement has been added to standard Ethernet switches to enable redundant rate-constrained communication with defined maximum latency. AFDX networks are used in commercial programs such as Airbus A380 and Boeing 787 as a backbone network for integrated avionics / IMA, and in military avionics (Airbus A400M) and rotorcraft cockpits for display integration. With a known network traffic profile, virtual link prioritization, defined switch buffer dimensioning and decent network calculus and configuration tools, very deterministic operation with respect to maximum latency can be ensured for all end-to-end virtual links (VLs) in AFDX networks. VLs enable point-to-point communication among different functions in the switched network system. Configured VLs have guaranteed bandwidth use and periodicity with defined maximum latency. ARINC664-P7 standard describes traffic policing and shaping required to ensure planned AFDX network performance.

Under assumptions of correct partitioning, airworthy internal network device architecture and bounded packet processing latency, which supports fault-tolerant distributed clock, the network design can be based on existing best practices for avionics network design associated with the ARINC 664 standard. The only difference is that synchronous networks rely on time-division and scheduling of the communication, while AFDX relies on statistical bandwidth partitioning and shaper configuration. All technologies for the design of critical systems can support partitioned virtual links (VLs) i.e. dataflows or “circuits” with defined temporal behavior.

From the perspective of the ARINC 664 network designer and integrator, additional synchronous communications in SAE AS6802 and IEEE TSN (Ref 11) messages can be seen as AFDX messages with fixed latency and jitter. While SAE AS6802 and ARINC664 can operate in one avionics network, additional mechanism would be required to support IEEE TSN traffic.

SUMMARY

The embedded system virtualization shall provide a full support for critical event-driven (signal/event/interrupts) and non-blocking periodic time-driven distributed computing to support two different paradigms in design of advanced control systems and functions. The embedded system virtualization shall create impression that all different functions are operating on private resources, and there are no conflicts with other functions on resource sharing. In an integrated system, this can be accomplished by using synchronous/asynchronous models of computation and communication (MoCC) (Ref 6) with system-level time-partitioning and Virtual Links (VLs) as a technology baseline. TTA MoCC ensures that distributed hard real-time functions can be hosted on an integrated embedded platform, while L-TTA and Relaxed TTA enable integration of time- and event-driven real-time control functions.

Therefore, the key technologies for integration of critical and non-critical functions in integrated architectures are (see Figure 5):

- Periodic TTA/L-TTA or TTA MoCC
- Time-Partitioning of computing resources if more than one critical function is hosted per CPU
- Congestion-free Ethernet communication and network bandwidth partitioning with defined QoS, based on deterministic dataflows / virtual links (i.e. synchronous/asynchronous or synchronous only links)

The listed technologies and MoCC support viable and sustainable design of robust critical infrastructure and non-critical soft-time function in one embedded networked system with tens of deterministic Ethernet switches and tens or hundreds of end stations (computers, remote IO, data concentrators, or smart sensors). All other thinkable mechanisms fail either on the costs or scalability side, or do not deliver expected performance, reliability and system robustness required for hosting of critical application in a mixed criticality environment.

If multicore processors are used and critical functions operate on those multicores then the communication among cores shall provide bandwidth partitioning with VL-like mechanisms supporting deterministic latency and jitter performance, unaffected by the Network-on-Chip (NoC) and multicore workload. This means that all used cores shall either have a direct physical access to the deterministic Ethernet network, or the NoC provides a comparable set of network virtualization mechanisms and bandwidth partitioning for all cores.

This opens new opportunities for system-level time partitioning for critical application, or more deterministic integration of less critical applications of media streaming, at different price point. Beyond protocol functions and services, it is essential to ensure that the networking devices are designed according to high safety assurance standards to avoid low-integrity behavior, as this reduces network reliability and availability. Also, the embedded platform shall pass the information on faults and health issues to application level for resolution. Both switch and end system are equally important to ensure high-integrity operation and prevent unintended behavior.

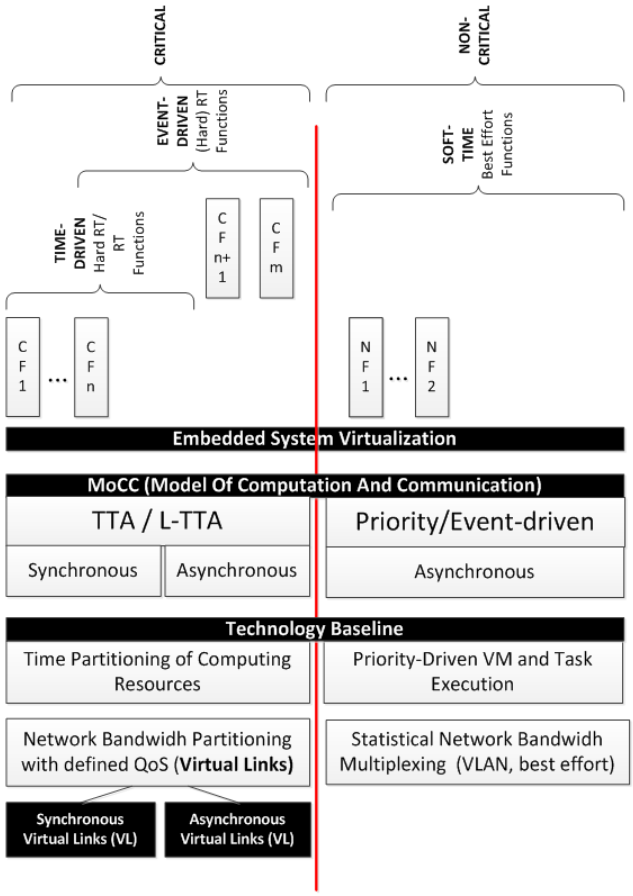

Figure 7. Switching Technology Baseline and MoCC for Advanced Integrated Architectures

INTEGRATING AUTONOMOUS OPERATION CAPABILITY

Autonomously driving cars must avoid obstacles with full understanding and awareness of surrounding environment and dynamics. This requires a quantum leap in terms of technology, system engineering and design methodology which is currently in progress. Huge investments by major automotive OEM and large consumer

companies is ongoing, and will lead to fully autonomous road vehicles by 2025-2030.

On rotorcraft platforms, the proposed set of solutions could integrate the standard avionics Ethernet-based architecture and the supplementary system for improved situation awareness, traffic collision avoidance, and autonomous operation. The reuse of standardized automotive components which are designed for deterministic real-time processing could provide new capabilities at a guaranteed safety level and much closer to automotive pricing. In addition, the described embedded platform can be an integral and natural part of the rotorcraft integrated architecture due to airworthy system integration capabilities and solid system and safety engineering tied to automotive standards ISO26262 ASIL-D. Internally such units already deploy resource partitioning like IMA.

The required embedded platforms should have significant processing power with tens of hundreds of CPU cores to capture all sensor data, enable deterministic processing as well as separation of functions. Typically, an automotive ADAS platform will host 50+ software functions with hard RT and soft time behavior.

INTEGRATING COTS HARDWARE FOR AUTONOMOUS OPERATION

The image shows how such a system ensembles very closely all aspects of IMA architectures (Figure 8) on one integrated unit (Figure 9) connected to the rotorcraft integrated architecture backbone, and how it can be extended to a fully integrated VPX system (Figure 10, Point B) as one of the common core components on the backplane.

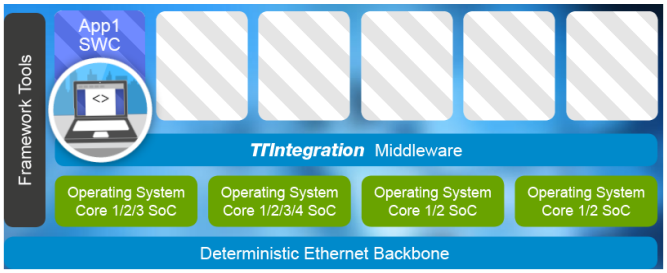

Figure 8. Internal ADAS architecture with capability of full integration as a common IMA component

The system architect can decide to attach a standard ECU unit with software adapted for rotorcraft applications on external unit, simply attached to the network. By defining interactions and sensor fusion approach which would bring sensor data to the ECU unit, full modularity is guaranteed, and the system operation of existing integrated rotorcraft avionics functions is not endangered at all. In a later phase, such hardware can be simply adapted to VPX module format, and plugged just as any other SBC into a core resource computer

Figure 9. Integrated architecture with different variants for ADAS embedded platform integration (ISO26262 ASIL-rated)

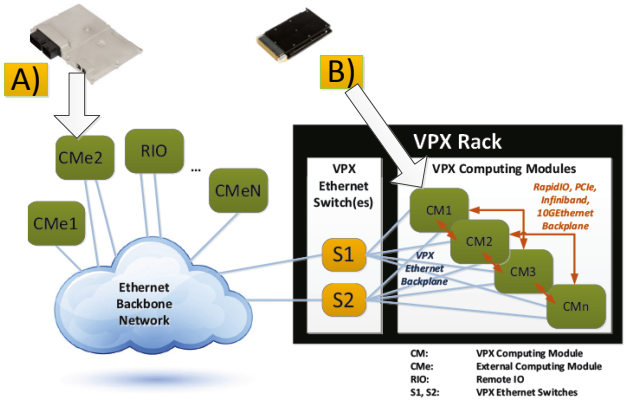

Figure 10. Logical View: Integrated architecture with different variants for ADAS embedded platform integration

CONCLUSIONS

The increasing presence of high reliability networks in automotive applications has narrowed the similarity gap with avionics networks. Meanwhile, the increased trend towards more connected complex aircraft networks creates a need to explore similar architectures in other fields.

Integrating the concept of heterogeneous Ethernet-based systems and topologies used in commercial integrated modular architecture (IMA) with integrated ADAS/ADS systems can yield a new generation of avionics architectures for rotorcraft applications.

There already exist tangible examples of network architectures in both industries gravitating towards open Ethernet platforms, such as Time-Triggered Ethernet. However, several factors will impact the transition to more similar topologies, such as commonality between certification standards, convergence in form factors, and economies of scale.

The proposed generic and open architecture based on deterministic Ethernet networking and VPX can enable a full integration of all sensors and control function on the aircraft. Per platform configuration, the system architect can determine which data is collected and essential for control system operation or IVHM.

Transitionally, the latest advances from other safety-relevant applications such as automotive ADS can be integrated either as external LRUs or adapted in VPX format, without any operational differences, due to deterministic high-bandwidth system integration suitable for commercial avionics.

Author contact:
Mirko Jakovljevic mirko.jakovljevic@tttech.com,
Alvaro Soares alvaro.soraes@tttechna.com

ACKNOWLEDGMENTS

The research leading to these results has received funding from the EU Horizon 2020 research and innovation program under grant agreement no 730830 and from the ARTEMIS JU under grant agreement n° 621429.

REFERENCES

[1] J. Gatard, M. Jakovljevic, "Evolution of Integrated Modular Architectures: Cross-Industry Perspective", Aviation Electronics Conference, March 2016, accessed online March 2017, http://www.ae-expo.eu/wp-content/uploads/2016/04/Jacques-Gatard-TTTech-Evolution_of_Integrated_Modular_Architectures_Cross-Industry_Perspective-FINAL.pdf

2 Takashi Yonemoto et al., "Engine Control ECU Development for '05 Standard Model", FUJITSU TEN TECHNICHAL JOURNAL, NO. 26 (2006), pp 3-9, accessed online March 2017, http://www.fujitsu-ten.com/business/technicaljournal/pdf/26-1.pdf

[3] IEEE802.3 website, Tutorial for Lifetime Requirements and Physical Testing of Automotive Electronic Control Units (ECUs), Accessed online http://grouper.ieee.org/groups/802/3/RTPGE/public/july12/hoganmuller_01a_0712.pdf

[4] SAE AS6802, Nov 2011, SAE Standards, http://standards.sae.org/as6802/

[5] ARINC664, "664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network",http://store.aviation-ia.com/cf/store/catalog_detail.cfm?item_id=1270, accessed Aug 2014

[6] Jakovljevic, M.; Ademaj A; "2nd Generation Ima: Extended Embedded Virtualization Capabilities For Optimized Architectures", Procedings of 32nd IEEE DASC, Oct 2013

[7] A. Benveniste, A. Bouillard and P. Caspi. A unifying view of Loosely Time-Triggered Architectures, EMSOFT '10 Proceedings of the tenth ACM international conference on Embedded Software, Oct 2010, Arizona, USA

[8] Kopetz H., Bauer G., The Time-Triggered Architecture, Proceedings of the IEEE, Volume 91, Issue 1, 2003

[9] Jakovljevic, M. Synchronous/asynchronous Ethernet networking for mixed criticality systems, Proceedings of IEEE/AIAA 28th Digital Avionics Systems Conference, 2009.

[10] Jakovljevic, M., Gatard J., "Deterministic high-bandwidth networks: Models of computation and system integration capabilities for actuator/sensor networks", Proceedings of International Conference on Recent Advances in Aerospace Actuation Systems and Components (R3ASC), April 2-3, 2014, Toulouse, France

[11] Julien Happich, "Getting started on IEEE's Time-Sensitive Networking", EETimes Europe, accessed March 2017, http://www.electronics-eetimes.com/news/getting-started-ieees-time-sensitive-networking